

APPENDIX A

1. A method for implementing a functional memory, in which memory data is stored as data units for each of which a dedicated storage space is assigned in the memory, in accordance with which method

- the memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different levels, wherein an individual node can be (i) a trie node associated with a logical table wherein an individual element may contain a pointer pointing to a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, in which case the content of the element corresponds to a nil pointer, the number of elements in the table corresponding to a power of two, or (ii) a bucket containing at least one element in such a way that the type of an individual element in the bucket is selected from a group including a data unit, a pointer to a stored data unit, a pointer to another directory structure and another directory structure,

- address computation performed in the directory structure comprises the steps of

- (a) selecting in the node at the uppermost level of the tree-shaped hierarchy a given number of bits from the bit string formed by the search keys employed, forming from the selected bits a search word with which the address of the next node is sought in the node, and proceeding to said node,

- (b) selecting a predetermined number of bits from the unselected bits in the bit string formed by the search keys employed and forming from the selected bits a search word with which the address of a further new node at a lower level is sought from the table of the node that has been accessed,

	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416	2417	2418	2419	2420	2421	2422	2
--	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	---

- repeating step (b) until an element containing a nil pointer is encountered

wherein the nodes to which a given node contains pointers are child nodes

characterized by

implementing trie nodes as quad nodes of four elements, and replacing in at

(a) an individual group comprising a given quad node and its child nodes is

(b) a compressed node known per se is formed from said node of 16

2. A method as claimed in claim 1, characterized in that

3. A method as claimed in claim 1, characterized in that

4. (Amended) A method as claimed in claim 2 [or claim 3],

characterized in that an upper limit is set for the number of pointers in the

APPENDIX A

compressed node, wherein when said limit is exceeded the compressed node is again decompressed to a quad node and child nodes.

5. A method as claimed in claim 4, characterized in that eight pointers is employed as said upper limit.

6. A method as claimed in claim 1, characterized in that ten pointers is employed as said upper limit.

7. (Amended) A method as claimed in claim 2 [or claim 3], characterized in that compression is additionally carried out on at least some of the quad nodes (N80...N82) in the structure in such a way that only non-nil pointers are physically stored in the node and in addition a bit pattern (BP2) on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.

8. A method as claimed in claim 1, characterized in that the non-nil pointers are stored in the compressed node in succession in the same order that they have in said table.

9. A method as claimed in claim 8, characterized in that the bit pattern has one bit for each element in the table, each bit indicating whether the corresponding element contains a nil pointer or a non-nil pointer.

10. A method as claimed in claim 8, characterized in that space is reserved for the bit pattern in all trie nodes of the directory structure.

11. A method as claimed in claim 8, characterized in that space is reserved for the bit pattern in the compressed nodes only.

APPENDIX A

12. A method for implementing a functional memory, in which memory data is stored as data units for each of which a dedicated storage space is assigned in the memory, in accordance with which method

- the memory is implemented as a directory structure comprising a tree-shaped hierarchy having nodes at several different hierarchy levels, wherein an individual node can be (i) an internal node associated with a logical table wherein an individual element may contain a pointer pointing to a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, in which case the content of the node corresponds to a nil pointer, the number of elements in the table corresponding to a power of two, or (ii) a leaf containing an element the type of which is selected from a group including a pointer to a stored data unit, a data unit, and a pointer to a node in another directory structure,

- address computation performed in the directory structure comprises the steps of

- (a) selecting in the node at the uppermost level of the tree-shaped hierarchy a given number of bits from the bit string formed by the search keys employed, forming from the selected bits a search word with which the address of the next node is sought in the node, and proceeding to said node,

- (b) selecting a given number of bits from the unselected bits in the bit string formed by the search keys employed, and forming from the selected bits a search word with which the address of a further new node at a lower level is sought from the table of the node that has been accessed,

APPENDIX A

- repeating step (b) until an empty element is encountered or until the address of the new node at a lower level is the address of a leaf,

wherein the nodes to which a given node contains pointers are child nodes of said given node and the nodes to which the child nodes contain pointers are grandchild nodes of said given node,

c h a r a c t e r i z e d b y

implementing internal nodes as quad nodes having four elements, and replacing in at least part of the directory structure groups of successive nodes by compressed nodes in such a way that

- an individual group comprising a given quad node and its child nodes is replaced by a node whose logical table has 16 elements, and

- a compressed node known per se is formed from said node of 16 elements by physically storing in the node only non-nil pointers and in addition a bit pattern on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.

13. A method as claimed in claim 12, c h a r a c t e r i z e d in that replacement is carried out in the directory structure on all groups in which the quad node has two child nodes.

14. A method as claimed in claim 12, c h a r a c t e r i z e d in that replacement is carried out in the directory structure on all groups in which the quad node has eight grandchild nodes at most.

15. A method as claimed in claim 13 [or claim 14], c h a r a c t e r i z e d in that an upper limit is set for the number of pointers in the compressed node, wherein

APPENDIX A

when said limit is exceeded the compressed node is again decompressed to a quad node and child nodes.

16. A method as claimed in claim 15, characterized in that eight pointers is employed as said upper limit.

17. A method as claimed in claim 15, characterized in that ten pointers is employed as said upper limit.

18. A method as claimed in claim 13 [or claim 14], characterized in that compression is additionally carried out on at least some of the quad nodes in the structure in such a way that only non-nil pointers are physically stored in the node and in addition a bit pattern (BP2) on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.

19. A method as claimed in claim 12, characterized in that the non-nil pointers are stored in the compressed node in succession in the same order that they have in said table.

20. A method as claimed in claim 19, characterized in that the bit pattern has one bit for each element in the table, each bit indicating whether the corresponding element contains a nil pointer or a non-nil pointer.

21. A method as claimed in claim 19, characterized in that space is reserved for the bit pattern in all trie nodes of the directory structure.

22. A method as claimed in claim 19, characterized in that space is reserved for the bit pattern in the compressed nodes only.

23. A memory arrangement for storing data units, said memory arrangement comprising a directory structure in which progress is made by using search words

APPENDIX A

formed from a bit string constituted by the search keys employed in each case, said directory structure comprising a tree-shaped hierarchy having nodes at several different hierarchy levels, wherein an individual node can be (i) a trie node associated with a logical table wherein an individual element may contain a pointer pointing to a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, in which case the content of the element corresponds to a nil pointer, the number of elements in the table corresponding to a power of two, or (ii) a bucket containing at least one element in such a way that the type of an individual element in the bucket is selected from a group including a data unit, a pointer to a stored data unit, a pointer to a node in another directory structure and another directory structure,

c h a r a c t e r i z e d in that

some of the trie nodes are quad nodes whose logical table has four elements and some are nodes whose logical table has 16 elements and in which only non-nil pointers are physically stored in addition to a bit pattern (BP1) on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.

24. A method as claimed in claim 23, c h a r a c t e r i z e d in that at least some of said quad nodes store physically only those pointers that are non-nil pointers and in addition a bit pattern (BP2) on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.

25. A memory arrangement for storing data units, said memory arrangement comprising a directory structure in which progress is made by using search words formed from a bit string constituted by the search keys employed in each case, said

APPENDIX A

directory structure comprising a tree-shaped hierarchy having nodes at several different hierarchy levels, wherein an individual node can be (i) an internal node associated with a logical table wherein an individual element may contain a pointer pointing to a lower node in the tree-shaped hierarchy and wherein an individual element may also be empty, in which case the content of the element corresponds to a nil pointer, the number of elements in the table corresponding to a power of two, or (ii) a leaf containing at least one element of a type selected from a group including a pointer to a stored data unit and a pointer to a node in another directory structure,

c h a r a c t e r i z e d in that

some of the trie nodes are quad nodes whose logical table has four elements and some are nodes whose logical table has 16 elements and in which only non-nil pointers are physically stored in addition to a bit pattern (BP1) on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.

26. A method as claimed in claim 23, c h a r a c t e r i z e d in that at least some of said quad nodes store physically only those pointers that are non-nil pointers and in addition a bit pattern (BP2) on the basis of which the physical storage location in the node, corresponding to the search word, can be determined.